DIGCOMP 3.4 Programming

To plan and develop a sequence of understandable instructions for a computing system to solve a given problem or perform a specific task.

KNOWLEDGE

K1: Knows that computer programs are made of instructions, written by humans according to strict rules in a programming language.

K2: Knows that programming languages provide structures that allow program instructions to be executed in sequence, repeatedly, or only under certain conditions, and to group them to define new instructions

K3: Knows that programs are executed by computing devices/systems, that are able to automatically interpret and execute instructions

K4: Knows that programs produce output data depending on input data, and that different inputs usually yield different outputs.

K5: Aware that for data (e.g. numbers, text, images, sounds) to be processed by a program, they have to be first properly digitized (i.e. digitally encoded).

K6: Knows that, to produce its output, a program stores and changes data in the computer system that executes it, and that it sometimes behaves unexpectedly (e.g. faulty behaviour, malfunction, data leakage)

K7: Knows that a program's blueprint is based on an algorithm, i.e. a step-wise method to produce an output from an input.

K8: Knows that various problem-solving strategies are used to develop algorithms.

K9: Knows that algorithms and consequently programs are intended to help solve real life problems; input data models the known information about the problem, and output data provides information relevant to the problem's solution. There are different algorithms and consequently programs solving the same problem.

K10: Knows that any program requires time and space (hardware resources) to compute its output, depending on the input's size and problem's complexity.

K11: Knows that there are problems that cannot be solved exactly by any known algorithm in reasonable time; thus, in practice they are frequently dealt with by approximate solutions (e.g. Artificial Intelligence applications).

SKILLS

S1: Able to write down precise instructions for another person to sort a deck of playing cards.

S2: Given a set of program blocks, being able to combine them to solve a problem.

S3: Knows how to detect issues in a sequence of instructions, and make changes to alleviate them (e.g. to find an error in the programme and correct it; find problems with execution time).

S4: Able to identify input and output data in some simple programs

ATTITUDES

A1: Willing to accept that algorithms, and hence programs, may not be perfect in solving the underlying problems.

A2: Developing the awareness that the responsibility of software failures should be attributed to the software developers

A3: Developing an appreciation for expressing actions and conditions precisely and unambiguously, even in natural language descriptions